Slide 0 – Title (≈ 30 sec)  
  
“Good morning everyone, I’m…, and I’m happy to present our work —   
Detecting Fraud in Real Time using Sequence-Based Feature Engineering and Modeling in Telecom.  
This focuses on transforming raw customer activity logs into structured, sequence-based features that help identify behavior patterns leading up to a decision event — whether that event turns out to be fraud or non-fraud.”  
  
Slide 1 – Title (≈ 30 sec)

Slide 2 – Business Motivation & Challenge  
  
In telecom, fraud is not a single action — it’s a series of activities over time.  
Traditional static models cannot capture this evolving behavior.  
We face three major challenges:  
1️⃣ High-volume data with millions of logs,  
2️⃣ Rare-event ratio — only 0.33% labeled as fraud,  
3️⃣ Noisy and overlapping activities between normal and suspicious behavior.  
  
The objective was to uncover meaningful patterns from these activity sequences and build explainable, real-time features that strengthen detection accuracy.”

Slide 3 – Data Source Summary (≈ 45 sec)  
  
Our data comes from the Telegence Channel, which logs every activity linked to a Billing Account Number (BAN).  
Each entry has a timestamp and a memo type that describes what action occurred.  
  
We began with nearly 250 memo types.  
Using a chi-square test with fraud and non-fraud labels, we identified the 13 most significant memo types — usually these are depicted in these forms ESCC, 1100, EQUIP, and CPCS.  
These memo types carried the strongest signals for differentiating normal versus anomalous behavior.”

Slide 4 – Pre-Processing Pipeline (48-Hour Window) (≈ 1 min 30 sec)  
  
The first step was to prepare structured input from unstructured logs.  
For every decision event, we used its unique Application ID and mapped all corresponding BAN activities that occurred within 48 hours before that event.  
  
The steps followed are:  
1️⃣ Filter all event logs for that BAN within the 48-hour window before the decision time,  
2️⃣ Retain only the top 13 memo types,  
3️⃣ Sort events chronologically,  
4️⃣ Compute the time gap between each event and the decision event in seconds.  
  
Now look at the slide:  
  
The top table shows decision event time @10 ‘0’ clock and the raw sequence— ESCC → NOTE → 1100 → CPCS with event times like 05:00, 07:00, 08:00, and 09:30.  
The bottom table shows the transformed version: sequence [ESCC, 1100, CPCS] with corresponding time gaps [10800, 5400, 1800] seconds.  
  
By pairing each event with its time gap, we capture not only what actions occurred but also when and how closely they happened — creating a clear behavioral timeline for every customer.”

**Slide 5 – Statistical Time-Gap & Sequence Features (≈ 1 min 40 sec)**  
From the **processed sequences** and their time gaps, we derived 5 key statistical features that summarize how active or **consistent** a customer’s recent behavior is.  
Each BAN We calculated:  
• Minimum gap – it is shortest interval between two events,  
• Maximum gap – longest pause,  
• Median gap – typical spacing,  
• Standard deviation gap – overall variability across actions,  
• Order length – number of events in the sequence.  
For the table, each BAN has its sequence and corresponding time gaps.  
  
For example:  
BAN\_101   
The median gap is 5400 seconds, and the standard deviation is 3699.  
For the BAN\_202, with standard deviation 2196.  
These statistics **reveal the tempo of user behavior.**  
When time gaps fluctuate heavily, the standard deviation gap increases — signaling irregular activity.  
When actions are evenly spaced or minimal, the variation remains small.  
This variability acts as a behavioral indicator that helps the model differentiate between steady and high-frequency activity patterns.

**Slide 6 – Token Rarity (IDF-Style Feature) (≈ 1 min 40 sec)**  
Next, we quantify how uncommon certain events are using token rarity — a concept similar to IDF in text analytics.  
The formula is:  
Rarity = log2(N/(1 + df))  
The upper table shows rarity scores for each memo:  
ESCC appears in 3 BANs – rarity 0.32,  
1100 appears in 3 BANs – rarity 0.32,  
CPCS appears in 1 BAN – rarity 1.00.  
CPCS is rarity event, so it gains high probability

The lower table combines these values for each sequence.  
For BAN\_101 → [ESCC, 1100, CPCS], the average token rarity = 0.55.  
This means the sequence includes both common and rare activities — a balanced but unique pattern.  
**Rarer activities** often stand out as **unusual** behaviors, and the model learns to treat them as signals that may indicate deviation from normal operational flow.”

**Slide 7 – Transition Probabilities & Sequence Likelihood (≈ 1 min 50 sec)**  
  
We also analyzed the transition flow between events — how one memo type moves to the next — to understand the behavioral order.  
Step 1: The top table lists raw transition counts.  
For example, ESCC → 1100 occurs 20 times, 1100 → CNTA 15 times.  
Step 2: The middle table shows the normalized conditional probabilities, such as:  
ESCC → 1100 = 0.44,  
ESCC → CNTA = 0.22.  
Step 3: The bottom table summarizes key sequence-level metrics:  
Probability Product = 0.044 (product of all the probabilities applied here conditional probability technique) i.e. next event given current event.  
Geometric Mean = 0.21,  
Entropy = 1.22.  
  
Entropy captures how **unpredictable the event order is.**  
A low entropy means stable and repetitive behavior; higher entropy **implies disorganized** and uncertain flow.  
This feature helps quantify the structure and consistency of user activity before the decision event.  
  
**Slide 8 – Bi-LSTM Autoencoder (≈ 1 min 50 sec)**  
  
To go beyond frequency and timing, we used an unsupervised Bi-LSTM Autoencoder to detect unseen patterns.  
This model learns the structure of normal sequences by **reading event orders both forward and backward.**It’s trained only on non-fraud sequences to establish what ‘normal’ looks like.  
  
During inference, a new sequence is **encoded and then reconstructed.**  
If reconstruction is good → low error (normal).  
If reconstruction fails → high error (abnormal).  
   
In the table:  
BAN\_101 → input and reconstructed match → error 0.00,  
BAN\_202 → slight deviation → 0.52,  
BAN\_303 → large mismatch → 0.71.  
The higher the error, the more the pattern diverges from learned normal behavior.  
This becomes a strong additional feature — an unsupervised anomaly score highlighting unusual activity even before the model labels it as fraud.”

**Slide 9 – Model Performance and Impact**  
  
After integrating these sequence-based features, we compared results against the baseline model.  
  
Baseline LightGBM (15 tabular features):  
Precision = 0.019, Recall = 52.6%, AUC = 0.896, AUC-PR = 0.048.  
  
Enhanced LightGBM (tabular + sequence features):  
After adding sequence-based features:  
Precision improved by 24 % (to 0.0235)  
Recall jumped by 37 % (to 72.3 %)  
AUC rose to 0.935 (+4.4 %)  
AUC-PR doubled to 0.096 , Earlier it was 0.048  
We also experimented with a Bi-LSTM + Conv1D hybrid model (ROC AUC 0.701, PR AUC 0.030), but the enhanced LightGBM offered a better trade-off between accuracy and speed.  
Overall, the addition of time-gap variability, rarity, transition flow, and anomaly-based features produced significant gains in recall and AUC for rare-event detection.

**“To conclude —**   
We transformed high-volume logs into structured 48-hour activity sequences for each BAN.  
We extracted timing, rarity, transition, and anomaly-based features — each giving a different behavioral view.  
  
Together, these sequence features improved interpretability and boosted model precision, recall, and overall AUC.  
This framework is now ready to scale into a real-time fraud detection pipeline.  
  
Thank you for listening — I’ll be happy to take your questions.”